WHAT IS CLAIMED IS:

1.   A PeerName managed class comprising a peer name string property field, an authority field, a classifier field, and a secured field.

2.   The PeerName managed class of claim 1, further comprising a first PeerName constructor to allow an application to specify the entire PeerName.

3.   The PeerName managed class of claim 1, further comprising a second PeerName constructor to allow an application to specify a PeerName based on peer identity information and a classifier to allow resolution of the PeerName to a third party.

4.   The PeerName managed class of claim 1, further comprising a third PeerName constructor to allow an application to specify a PeerName based on a PeerName object and a classifier to allow resolution of the PeerName to a third party.

5.   The PeerName managed class of claim 1, further comprising at least one method exposed thereby selected from the group consisting of an equals method, = = method, a GetHashCode method, a GetType method, a ReferenceEquals method, and a ToString method.

6.   A PeerIdentity managed class comprising a PeerName field, a FriendlyName field, and a Key field.

7.   The PeerIdentity managed class of claim 6, further comprising a first constructor to retrieve an identity based on a PeerName parameter.

8.   The PeerIdentity managed class of claim 6, further comprising a second constructor to create a new identity based on a friendlyName parameter, a classifier, and a key.

9.   The PeerIdentity managed class of claim 6, further comprising a third constructor to create a new identity based on friendlyName parameter and a classifier.

10.   The PeerIdentity managed class of claim 6, further comprising a fourth constructor to create a new identity based on a friendlyName parameter, a blank classifier, and a new key pair.

11.   The PeerIdentity managed class of claim 6, further comprising at least one method exposed thereby selected from the group consisting of an export method, a CreatePeerName method, a GetInfo method, a delete method, an equals method, a GetHashCode method, a GetType method, a ReferenceEquals method, and a ToString method.

12.   The PeerIdentity managed class of claim 6, further comprising at least one static method exposed thereby selected from the group consisting of a GetIdentities static method, a first import static method utilizing an ExportedPeerIdentity and a password to import a PeerIdentity, and a second import static method utilizing an ExportedIdentityXxl string and a password to import a PeerIdentity.

13.   A PeerIdentityInfo managed class comprising a PeerName field, a FriendlyName field, and a Key field.

14.   The PeerIdentityInfo managed class of claim 13, further comprising a PeerIdentityInfo constructor that constructs a PeerIdentityInfo object from an IdentityInfoXml string.

15.   The PeerIdentityInfo managed class of claim 13, further comprising at least one method exposed thereby selected from the group consisting of a ToXmlString method, an equals method, a GetHashCode method, a GetType method, a ReferenceEquals method, a GetObjectData method, and a ToString method.

16. An ExportedPeerIdentity managed class comprising an ExportedPeerIdentity constructor that utilizes an ExportedIdentityXmlString to construct an ExportedPeerIdentity object.

17. The ExportedPeerIdentity managed class of claim 16, further comprising a PeerName property that returns the PeerName of the exported identity.

18. The ExportedPeerIdentity managed class of claim 16, further comprising at least one method exposed thereby selected from the group consisting of a ToXmlString method, an equals method, a GetHashCode method, a GetType method, a ReferenceEquals method, a GetObjectData method, and a ToString method.

19. A method of managing by an application a PeerName in a managed framework, the method comprising the steps of:

communicating with a managed PeerName object, the managed PeerName object exposing constructors for creating an entire PeerName object, creating a hypothesized PeerName object for resolution to a third party from PeerIdentityInfo and a classifier, and creating a second hypothesized PeerName object for resolution from a known PeerName object;

selecting one of the constructors;

passing to the managed PeerName object parameters required by the constructor selected; and

initiating the constructor.

20. A method of managing by an application a PeerIdentity in a managed framework, the method comprising the steps of:

communicating with a managed PeerIdentity object, the managed PeerIdentity object exposing constructors for creating a new identity from parameters of FriendlyName, classifier, and key, for creating a new identity from parameters of FriendlyName and classifier, for creating a new identity from parameters of FriendlyName, a blank classifier, and a new key pair, and retrieving an identity associated with a given PeerName parameter;

selecting one of the constructors;

passing to the managed PeerIdentity object parameters required by the constructor selected; and

initiating the constructor.

21. The method of claim 20, wherein the managed PeerIdentity object further exposes static methods for importing a peer identity from an ExportedPeerIdentity and a password, for importing a peer identity from an ExportedIdentityXml string and a password, and for retrieving a collection of identities for a user account, the method further comprising the steps of selecting one of the static methods, passing to the managed PeerIdentity object parameters required by the static method, and initiating the static method.

22. A method of managing by an application PeerIdentityInfo in a managed framework, the method comprising the steps of:

communicating with a managed PeerIdentityInfo object, the managed PeerIdentityInfo object exposing a constructor for creating a PeerIdentityInfo object from an IdentityInfoXml parameter;

passing to the managed PeerIdentityInfo object the parameter required by the constructor; and

initiating the constructor.

23. A method of managing by an application a ExportedPeerIdentity in a managed framework, the method comprising the steps of:

communicating with a managed ExportedPeerIdentity object, the managed ExportedPeerIdentity object exposing a constructor for creating an ExportedPeerIdentity object from an ExportedIdentityXmlString;

passing to the managed ExportedPeerIdentity object parameters required by the constructor; and

initiating the constructor.